



slip angle  $\theta_s$  can be calculated as the time integral of the slip angle velocity  $\omega_s$ , by adding the rotor angle  $\theta_r$  to the slip angle the rotor flux position  $\theta_m$ , many be calculated:

$$\theta_s = \theta_r + \theta_m \quad (1)$$

The mathematical model of induction motor applied in the paper, has been obtained after the transformation of the stator and the rotor phase equations into two, rotating with synchronous velocity  $\omega_s$  orthogonal axes.

$$\left\{ \begin{aligned} \frac{d}{dt} i_{sd} &= \frac{1}{\sigma \cdot L_s} \left[ -\left(R_s + \frac{M^2 R_r}{L_r^2}\right) \cdot i_{sd} + \omega_s \cdot \sigma \cdot L_s \cdot i_{sq} + \frac{MR_r}{L_r^2} \cdot \psi_{rd} \right. \\ &\quad \left. + \frac{M}{L_r} \cdot \omega_r \cdot \psi_{rq} + u_{sd} \right] \\ \frac{d}{dt} i_{sq} &= \frac{1}{\sigma \cdot L_s} \left[ -\omega_s \cdot \sigma \cdot L_s \cdot i_{sd} - \left(R_s + \frac{M^2}{L_r \cdot T_r}\right) \cdot i_{sq} - \frac{M}{L_r} \omega_r \cdot \psi_{rd} \right. \\ &\quad \left. + \frac{M}{L_r \cdot T_r} \cdot \psi_{rq} + u_{sq} \right] \end{aligned} \right. \quad (2)$$

$$\left\{ \begin{aligned} \frac{d}{dt} \psi_{rd} &= \frac{MR_r}{L_r} \cdot i_{sd} - \frac{R_r}{L_r} \cdot \psi_{rd} + (\omega_s - \omega_r) \cdot \psi_{rq} \\ \frac{d}{dt} \psi_{rq} &= \frac{MR_r}{L_r} \cdot i_{sq} - (\omega_s - \omega_r) \cdot \psi_{rd} - \frac{R_r}{L_r} \cdot \psi_{rq} \end{aligned} \right. \quad (3)$$

$$\sigma = 1 - \frac{M^2}{L_s L_r}$$

where  $i, u, \psi$  denote current, voltage and flux linkage respectively. Subscripts r and s stand for rotor and stator.  $\omega_r$  is the rotor speed,  $L$  and  $R$  are the auto-inductances and resistances,  $M$  is the mutual inductance.  $\sigma$  constant coefficient. The motor load system can be described by a fundamental torque equation (4):

$$T = T_l + J \frac{d\Omega_m}{dt} + f\Omega_m = \frac{n_p M}{L_r} (\psi_{rd} i_{sq} - \psi_{rq} i_{sd}) \quad (4)$$

where  $T$  the instantaneous value of the developed motor torque.  $T_l$  the instantaneous value of the load torque.  $\Omega_m$  rotor speed of the motor shaft.  $J$  the moment of inertia of the motor load system.  $F$  the coefficient of friction the number of pair poles.

### B. Indirect Vector Control of an Induction Motor

Based on reference frame theory, the induction motor drive can be controlled like a separately excited dc machine by field oriented control method [1], which can be design in tow basic ways by direct or indirect method. The choice between these two methods is not obvious because each method has its distinctive advantages and disadvantages. As a result a great research effort has been made to improve both direct and indirect field oriented controllers by design of complicated hardware and software to compensate non-ideal machine behaviour such as parameter variations due to temperature changes, rotor deep bar effects and magnetic saturation. The bloc diagram shown in Fig. 2, depicts the general structure of

the indirect field oriented control with speed control motor drive, has been chosen for control of induction motor drive.

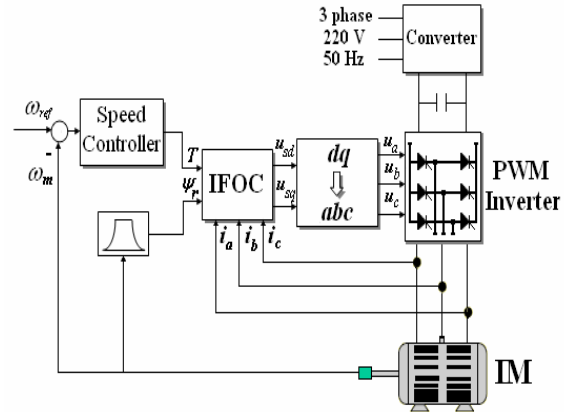


Fig. 2 Indirect field oriented induction motor drive

This scheme includes induction motor, pulse with modulated (PWM) inverter, indirect field oriented control (IFOC) and speed controller. In this approach the flux angle  $\theta_s$  is not measured directly, but is estimated from the equivalent circuit model and from measurements of the rotor speed, stator currents, and the voltages  $u_{sd}, u_{sq}$ .

### C. PI Speed Controller

The dynamic model of speed induction motor drive is significantly simplified, and can be reasonably represented by the bloc diagram shown in Fig. 3.

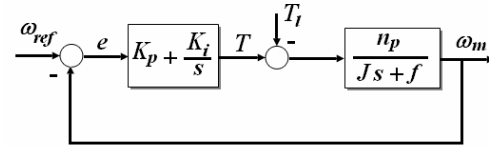


Fig. 3 Bloc diagram of speed system controller

By using the Laplace transformation, the transfer function for (4) is equation (5):

$$\omega_m(s) = \frac{n_p(T - T_l)}{Js + f} \quad (5)$$

The classic numerical PI (Proportional and Integral) regulator is well suited to regulating the torque, to the desired values as it is able to reach constant reference, by correctly both the P term ( $K_p$ ) and I term ( $K_i$ ) winches are respectively responsible for error e sensibility and for the steady state error. If  $T_l = 0$ , the transfer function is as following (6):

$$G(s) = \frac{n_p(K_p s + K_i)}{Js^2 + (f + K_p n_p)s + K_i n_p} \quad (6)$$

where

$$P(s) = s^2 + \frac{f + K_p n_p}{J} s + \frac{K_i n_p}{J} = 0 \quad (7)$$

The expressions for  $K_p$  and  $K_i$  of the regulator is calculated by Imposition of poles complexes combined with real part negative  $S_{1,2} = \rho(-1 \pm j)$ .

$$\begin{cases} K_p = \frac{2\rho J - f}{n_p} \\ K_i = \frac{2J\rho^2}{n_p} \end{cases} \quad (8)$$

where  $\rho$  It is a positive constant.

The proposed indirect vector control has several advantages over conventional one as are its independence of the motor model parameters and simple microcomputer implementation. The effects of stator resistance  $R_r$  variations in the calculation of slip frequency and transformation angle is compensated by motion controller. The nonlinearities caused by magnetic saturation can be compensated by the inverse magnetizing characteristic.

### III. CONTROL SYSTEM BASED ARTIFICIAL NEURAL NETWORK

#### A. Adaptive Neural Networks

A general architecture graph of Multilayer Perceptron (MLP) is shown in Fig. 4. This network which can be multiplexed for each output of the controller has been found to possess acceptable performance in many industrial applications. The feed-forward topology shown in the network of Fig. 4 offers the advantage of simplicity and ease programming. Such a neural network contains three layer, hidden layers and output layer. Each layer is composed of several neurons. The number of the neurons in the output and layers depends on the number of the selected input and output variables. The number of hidden layers and the number of neurons in each depend on the system dynamic and the desired degree of accuracy.

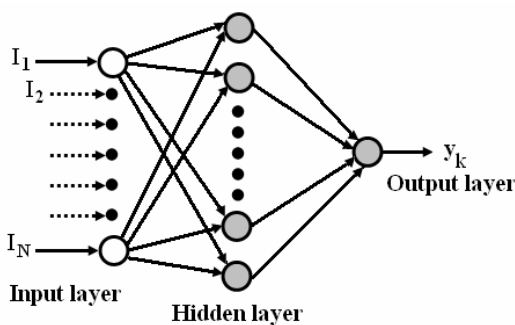


Fig. 4 Architecture of Multilayer Neural Network

The block-diagram in Fig. 5 shows the model of a neuron, which performs two functions. The first is to sum all the inputs from the upper layer based on their weighting factors in equation (9). The second is to process this sum by a nonlinear sigmoid function in equation (10).

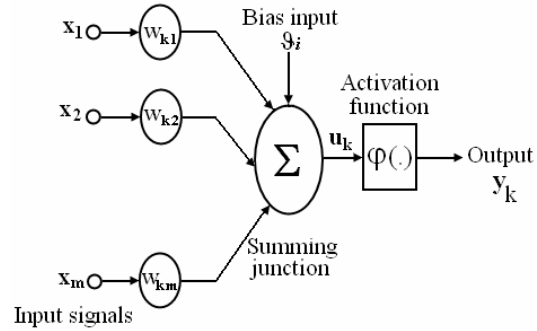


Fig. 5 Basic model of formal neurone

The basic equations describing the dynamics of each neuron are

$$u_k = \sum_{j=1}^m w_{ji} x_j \quad (9)$$

$$y_k = \phi(u_j + v_j) \quad (10)$$

where  $w_{ji}$  design the synaptic weight between the  $j$ th neuron and the  $i$ th neuron in two adjacent layers.  $\phi(\cdot)$  is the activation function. The neural network has two phases of operations, training and testing. In the training phase, the weights of the neural network are adjusted to map the input of the system to its output. In the testing phase, the neural network should predict the correct system output for a given input, even if the input was not used in training. Here for generality, the scalar weighted summing of the input array  $x_i$  is distorted by a linear function  $\phi(\cdot)$  which is usually sigmoidal (e.g. tanh function) to facilitate the gradient search techniques used in the training procedure. An Adaptive Neural Networks (ANN) is made up of many such neurons arranged in a variety of architectures. The feed-forward architecture graph shown in Fig. 4 offers the advantage of simplicity and ease of programming.

#### B. Training Neural Network

The most common method of neural network is error back-propagation algorithm [7]. The algorithm is based on the gradient descent search technique that minimizes a cost function of the mean square errors. The minimization process is done by adjusting the weighting vector of the neural network. Several training algorithms have been proposed to adjust the weight values in dynamic recurrent neural network. Examples for these methods are the dynamic back-propagation from Narendra and Patthasarathy; 1991; Williams and Ziepser, 1995; among others. The cost function being minimized is the error between the network output and the desired output given by equation (11):

$$E = \frac{1}{2} \sum_j e_j^2(k) = \frac{1}{2} \sum_j [y_j^* - y_j(k)]^2 \quad (11)$$

where  $y_j(k)$  is the output of neuron  $j$  and  $y_j^*(k)$  is the desired pattern for that neuron. Let  $\eta_{ji}(k)$  denote the learning

rate parameter assigned to synaptic weight  $w_{ji(k)}$  at iteration number  $k$ . Minimizing equation (12) leads to a sequence of update of weight vector. The weights of the interconnections between two adjacent layers can be update based on the following formula:

$$w_{ji}(k+1) = w_{ji}(k) - \eta_{ji}(k+1) \frac{\partial E(k, w)}{\partial w_{ji}(k)} + \alpha \Delta w_{ji}(k) \quad (12)$$

$\alpha$  is the momentum gain, is susceptible to local minima and needs additional computation for gradient evaluation, and  $\Delta w_{ji}(k)$  is weight change based on gradient of the cost function  $E_{k,w}$  and  $k$  is the iteration number.

### C. Adaptation Learning Control Scheme

The proposed adaptive neural network controller is shown in Fig. 6 where as the structure of the neural network used is featured in Fig. 7. In off line training the targets are provided by an existing controller, the neural network adjusts its weights until output from the ANN is similar to the controller.

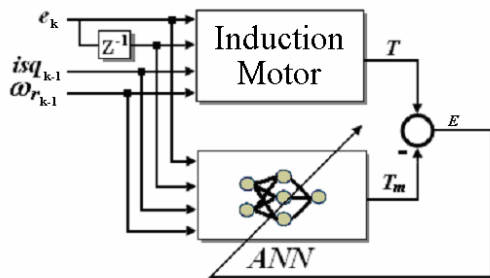


Fig. 6 Supervised learning using an existing controller

The four input signals ( $e(k)$ ,  $e(k-1)$ ,  $isq(k-1)$ ,  $w_r(k-1)$ ), and the torque ( $T(k)$ ) output are exported to the MATLAB Workspace. The following MATLAB code trains the Neural Network. The first section of code generates the 'cell array'. The cell array combines the 4 different inputs into 1 input vector. The Feed-forward network has 10 neurons in the two layers Fig. 7. The activation function in the two layers is tan-sigmoid and the output layer is a linear function.

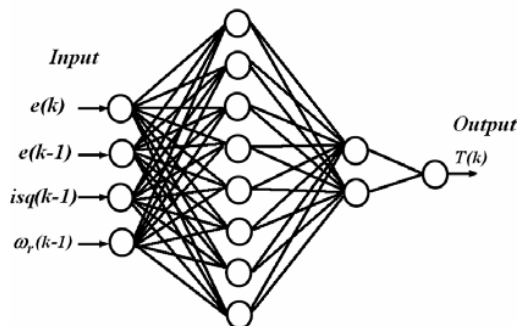


Fig. 7 Multilayer Feedforward Neural Network

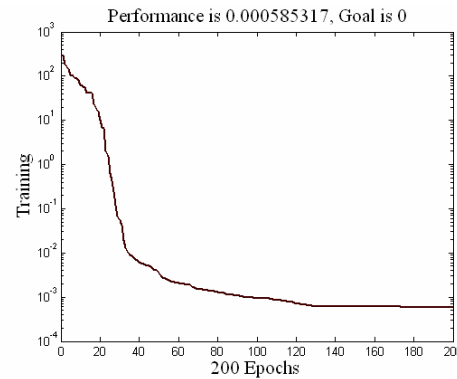


Fig. 8 Training error

Where the training is finished, the weights are set Fig. 8, and a Simulink ANN is generated. The network is placed in the existing PI controller Fig. 9, in the block diagram of indirect oriented Field vector controlled.

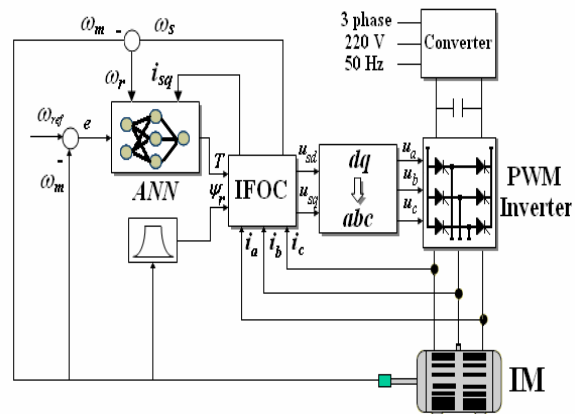


Fig. 9 Vector drive with ANN based feedback signal processing

## IV. RESULTS AND DISCUSSION

The vector controller induction motor drive using artificial neural network was simulated through Matlab-Software with Simulink Toolboxes. The simulation phase was very important to verify correctness of theoretical assumptions and to find behaviour of the drive. The parameters values of the system under study are summarized in Table I.

TABLE I  
RATING OF TESTED INDUCTION MOTOR

| Rated values     | Power          | 1.5       | kW  |
|------------------|----------------|-----------|-----|
|                  | Frequency      | 50        | Hz  |
|                  | Voltage Δ/Y    | 220/380   | V   |
|                  | Current Δ/Y    | 11.25/6.5 | A   |
|                  | Motor Speed    | 1420      | rpm |
|                  | pole pair (p)  | 2         |     |
| Rated parameters | R <sub>s</sub> | 4.85      | Ω   |
|                  | R <sub>r</sub> | 3.805     | Ω   |
|                  | L <sub>s</sub> | 0,274     | H   |
|                  | L <sub>r</sub> | 0,274     | H   |

|           |     |         |                   |
|-----------|-----|---------|-------------------|
|           | M   | 0,258   | H                 |
| Constants | J   | 0,031   | Kg.m <sup>2</sup> |
|           | $f$ | 0,00114 | Kg.m/sec          |

The proportional and derivative parameters of the proposed control scheme are  $K_p=0.58$  and  $K_i=11.19$ . Several test cases were considered in order to evaluate the performances under a variety of operating conditions. For the robustness of the proposed control scheme, we assure that the parameters of rotor resistance  $R_r$  and load inertia  $J$  have been perturbed from their nominal values Figs. 10, 11. The parameters of the stator resistance, inductance and viscous friction change their nominal values. It is evident that the speed response of the proposed control scheme is not significantly affected by this variation. One can see from these all figures the results were very successful and the obtained results confirm the validity of the proposed control scheme.

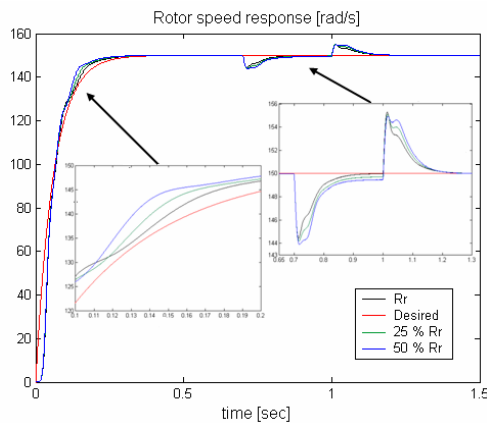


Fig. 10 Results of speed evolution after resistance changes

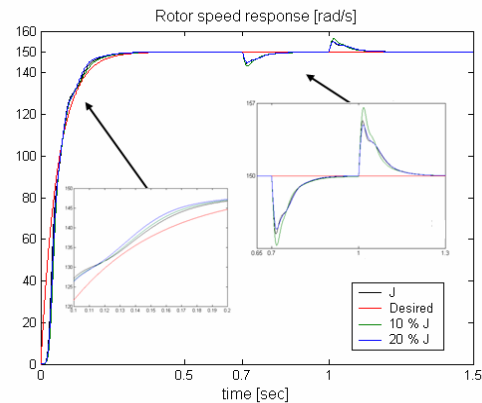


Fig. 11 Results of speed evolution after inertia load changes

The Fig. 12 shows the behaviour of the system to screw of resistant torque  $T_r=10 \text{ N.m}$ , his disturbance can be seen at  $t=0.5 \text{ s}$  and  $t=2 \text{ s}$ , in maintaining the constant speed control  $\omega_r=150 \text{ rad/s}$  and  $\omega_r=-150 \text{ rad/s}$  at  $t=1.5 \text{ s}$ . Fig. 13 shows the results by Reference without filter, the results were very successful and the obtained results confirm the validity of the proposed controller. To demonstrate the robustness of the

proposed controller, Fig. 14 displays the results of speed control using neural controller with stochastic lead change, the neural controller reduces both the overshoot and extent of oscillation under the same separating condition.

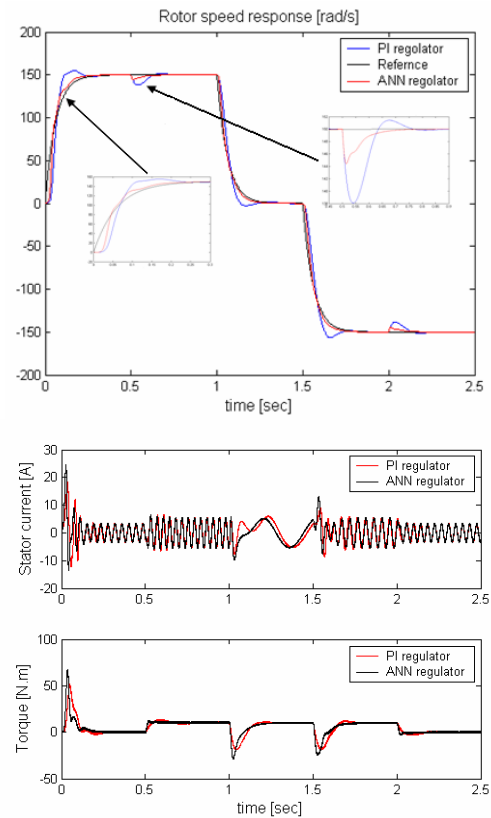
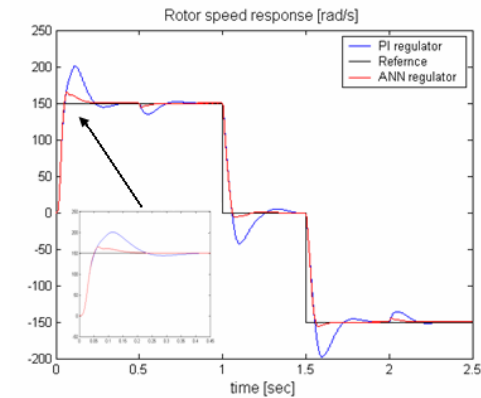


Fig. 12 Speed control system using neural controller and PI controller



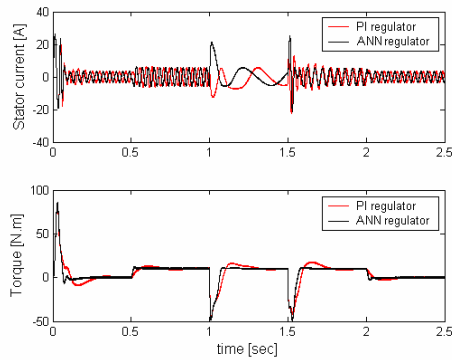


Fig. 13 Speed control system using neural controller and PI controller

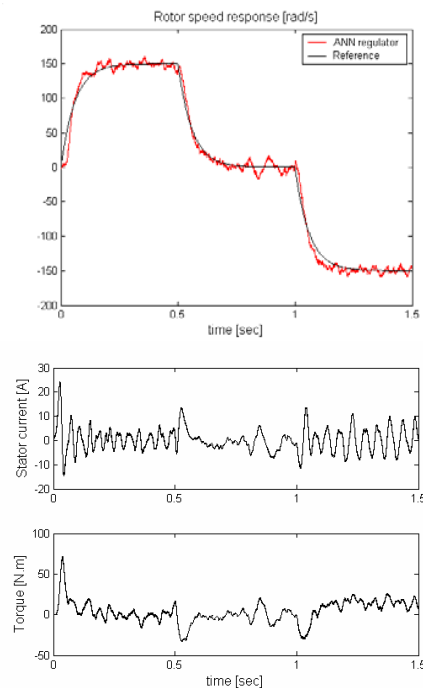


Fig. 14 Speed control using neural controller with stochastic lead change

## V. CONCLUDING REMARKS

In this paper, we have designed and implemented the neural network controller NNC, for accurate speed control of an induction motor. Comparing PI-type control method, simulation results are provided to illustrate the performance and the effectiveness of the proposed control scheme, even in the presence of much strong mechanical friction and other non linear characteristics.

The success of the designed controller is demonstrated in real-time under load conditions by applying a load torque to the shaft of the motor. The results show that the controller could compensate for this kind of disturbances. The plant is also tested for the tracking property using different types of reference signals. Satisfactory performance was observed for most reference tracks and the results demonstrated the effectiveness of the proposed structure and the proposed control scheme it is believed will constitute a major step in the

evolution of intelligent control of complex mechatronic systems.

## REFERENCES

- [1] C.M.Liaw, Ys.Kung and C.M.Wa, Design and implementation of a high performance field-oriented induction motor drive. IEEE Trans. Ind. Electron. Vol 38,4, pp 275-282, 1991.
- [2] Jie Zhang and T.H.Burton, New approach to field orientation control of CSI induction motor drive. IEE Proceedings, Vol 135, No 1, January 1988.
- [3] P.Pillay and R.Krishnam, Modelling of permanent magnet motor drive. IEEE Trans. Ind. Electronics. Vol 35, No 4, pp 537-541, 1988.
- [4] Tien Chi Chen and Tsong Terng Shen, Model reference neural network controller for induction motor speed. IEEE Trans energy conversion. Vol 17 No 2, pp 3301-3305, 2004.
- [5] T.C.Chen and T.T.Sheu, Robust speed controlled induction motor drive based on model reference with neural networks. Inter. Journ of Knowledge based Intelligent Engineering System. Vol 3.3 pp 162- 171, 1992.
- [6] K.S.Narendra, Neural networks for control: Theory and practice. Proceedings of the IEEE, 84,10 pp 1385-1406, 1996.
- [7] M.Kuchar, P.Brandstetter, M.Kaduch, Sensorless, Induction motor drive with neural network 35th conference pp 3301-3305, 2004.
- [8] Y.A.Kwon and S.Hwankin, A new scheme for speed sensorless control of induction motor. IEEE Trans Ind Electronics, Vol 51 No3, pp 545-550, 2004.
- [9] W.S.Oh, K.M.Cho, S.Kun and H.J.Kim, Optimized neural network speed control of induction motor using genetic algorithm. IEEE International Symposium on Power Electronic, Electrical drives, Automation and Motion, pp 12-15, SPEEDAM 2006.
- [10] N.Bongkyong, L.Hjoon and P.Sanglong, A multiobjective evolutionary neuro-controller for non-minimum phases system. IEICE Trans INF & Syst Vol.E87 D, No11 pp 2517-2520, 2004.



**Mokhtar Zerikat** (M'05) received the B.S degree in electrical engineering and the M.S. and Ph.D degrees in electronics from the University of Sciences and Technology of Oran, Algeria, in 1982, 1992, and 2002, respectively. Following graduation, he joined the university as a lecturer. He is currently a professor in the department of electrotechnics engineering. He is engaged in research and in education in the area of electrotechnics. Actually he is associate Professor at Electrical Engineering Institute (ENSET) of Oran-Algeria. His current research includes electrical machines, High-performance motor drives, modeling and adaptive control systems, image processing and development intelligent applications. He is actively involved in many projects with industry while engaged in teaching, research and consulting in the area of artificial intelligence. Actually, he is associate Professor at the department of electrical engineering (ENSET) of Oran in Algeria. He has authored more than 45 technical papers published in international journals and conferences proceedings. He authored a book on automatic Control, which was recently published and regularly gives tutorials and industrial courses on these subjects.

Dr. Zerikat is a chairman of the institute of electrical and electronic engineers in the Algeria state. He is regularly reviews papers for several journals in his area. He is an active member of the industrial COST Education Committee.



**Soufyane Chekroun** was born in Tlemcen, Algeria, in 1983. He received his B.S and M.S degrees in Electrical Engineering from the University of Sciences and Technology of Oran, Algeria. Following graduation, he joined the Institute of Electrical Engineering of ENSET Oran, Algeria, as a research associate.

His research interests include Electrical machines and development of intelligent applications (fuzzy logic and neural networks) and the practical developments of high performance electrical motor drives and their related knowledge control schemes.